audiokinetic

Installation and Migration Guide

Wwise® 2019.1.1

Installation and Migration Guide

Wwise® 2019.1.1

Table of Contents

Chapter 1. Welcome to Wwise

Introducing Wwise

Welcome to Wwise, the middleware solution from Audiokinetic that gives you the power to create great audio and motion for video games. Through the tight integration of an advanced authoring application and a robust sound engine, Wwise can increase your productivity and enhance your creative output.

Built to address the specific needs of the game development pipeline, Wwise is a unique solution for designers, composers, scriptwriters, and programmers. By allowing you to develop game audio, music, dialogue, and motion concurrently with game visuals, Wwise facilitates the design and authoring of sophisticated audio and motion during every phase of game development.

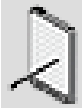
About This Guide

This guide is designed for audio and motion designers, integrators, composers, scriptwriters, and programmers working in the game development industry who may need to install and upgrade the Wwise authoring application, the Wwise SDK, and additional Wwise components. Some basic concepts are explained, but some sections assume familiarity with code modifications. This guide gives you specific information and step-by-step instructions to install and upgrade the Wwise Launcher and, in turn, all other Wwise components.

Chapter 2. Installing Wwise and its Components

Wwise System Requirements

Before you install Wwise, you should verify its minimum system requirements. The following table describes the hardware and software requirements to install and run the Wwise authoring application.

System Component	Requirement
Processor	The following processors are required to run Wwise: • Pentium 4® Processor 3.0 GHz or better with Hyper-Threading technology • AMD Athlon 3.0 GHz or better
Operating System	The following operating systems are supported: • Windows 7 with SP 1 or more recent • Mac OS X Yosemite (10.10) or more recent  **Note** For the Wwise sound engine alone, Mac OS X Mavericks (10.9) is also supported. The Wwise authoring application is only available in 64 bits.
Memory	1 GB of RAM  **Note** Large projects require larger amounts of RAM. Use the 64-bit version of the Wwise authoring application to use more than 3 GB of RAM.
Resolution	The factory layouts have been optimized for monitors with a resolution of 1920 x 1080, but they can be adapted for other resolutions. Support for 4k or 5k is available by enabling the **Enable High DPI support** User Preferences option.
PDF Reader	A reasonably recent version of Adobe Reader or other PDF reader is required to open the PDF files that are distributed with Wwise.
QuickTime	AAC conversion in Wwise for the Mac® and iOS platforms requires QuickTime 7.6.5 or later.
Mandatory Code Dependencies	
Microsoft .NET Framework	Microsoft .NET Framework v4. The Wwise Launcher will, as needed, automatically install this component before installing a version of Wwise. If, however, you deploy Wwise without using the Launcher (via Perforce or some other revision control system), in order to run the File Packager you will need to install the .NET Framework manually using Microsoft .NET 4 (Web Installer) from the Microsoft website.
Microsoft DirectX®	DirectX June 2010 runtime. The Launcher will automatically detect if this external component is on your system and, if not, will install it before installing Wwise. If, however, you deploy Wwise without using the Launcher (via Perforce or some other revision control system), you will need to update the DirectX runtime manually. You can do that by downloading it directly from the Microsoft Download Center .

System Component	Requirement
	 **Note** Without the appropriate DirectX version on your computer, Wwise will run normally; but, the Game Object 3D Viewer will not be available and you may not be able to test Motion in Wwise.
Microsoft Visual Studio Redistributable	The following redistributables must be installed to run the Wwise authoring tool: • Microsoft Visual C++ Redistributable Package for Visual Studio 2013 (64-bit) • Microsoft Visual C++ Redistributable Package for Visual Studio 2015 (64-bit) The Launcher will automatically detect if these are on your system and, if not, will install them before installing Wwise.

Be Sure to Install All the Mandatory Code Dependencies

It is not uncommon for a crash to occur due to one or multiple missing dependencies when installing Wwise without the Launcher. The required dependencies are detailed in the following rows of the Wwise System Requirements table: Microsoft .NET Framework, Microsoft DirectX, and Microsoft Visual Studio Redistributable. Installing the Authoring tool through the Launcher is less troublesome because you don't have to worry about the dependencies since they are automatically provided.

The Wwise Launcher

Wwise, its tools, samples, integrations, and other components are installed, uninstalled, and opened using the Wwise Launcher, hereafter referred to simply as the Launcher.

To install the Launcher:

1. Click **Download Wwise** from the Audiokinetic website's download page.
2. On Windows: Run the downloaded *WwiseLauncher.msi*.
   On Mac: Run the downloaded *WwiseLauncher.pck*.

After you have downloaded and installed the Launcher, occasional prompts will request to install updates. Although you will not have to run them immediately, it is highly recommended to stay up to date. They are quick and will not impact your Wwise installations and integrations.

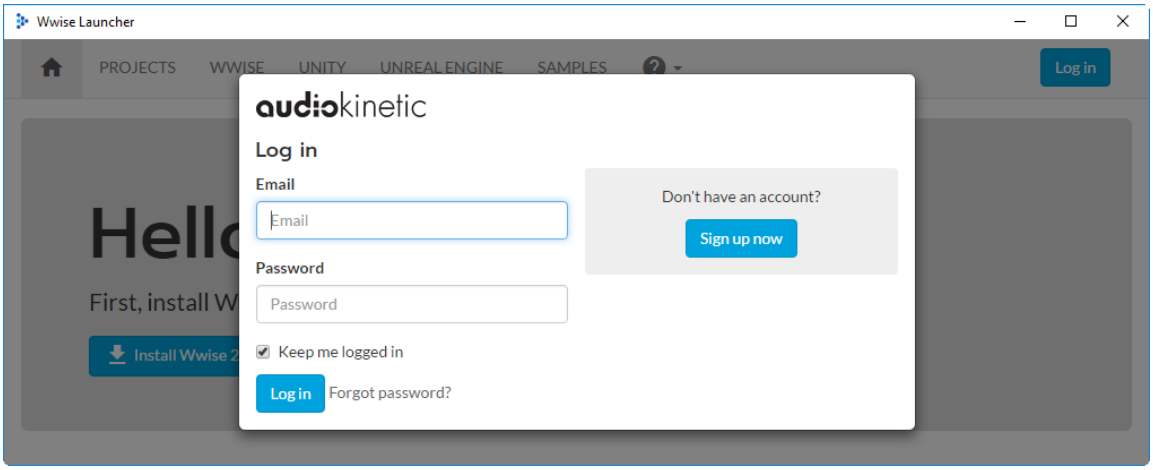
Wwise Launcher System Requirements

Like Wwise, the Launcher works on Windows and Mac. And, if you have the necessary Wwise System Requirements, you will have all you need for the Launcher in terms of memory, RAM, and processing power.

You will, however, need a stable Internet connection. Without one, the Launcher will still be able to start already-installed Wwise versions and their components; but, installing new versions or components, retrieving news and community content in the Home tab, and connecting to your customer portal will not be possible.

First-time Use

If this is the first time installing the Launcher, it will open with a prompt to log in, as can be seen in the screenshot below.

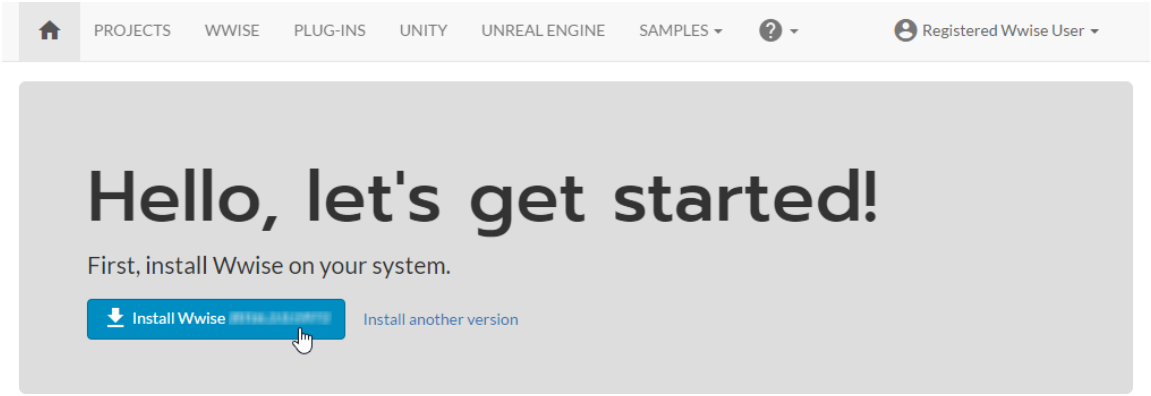


1. Log in using your registered account.

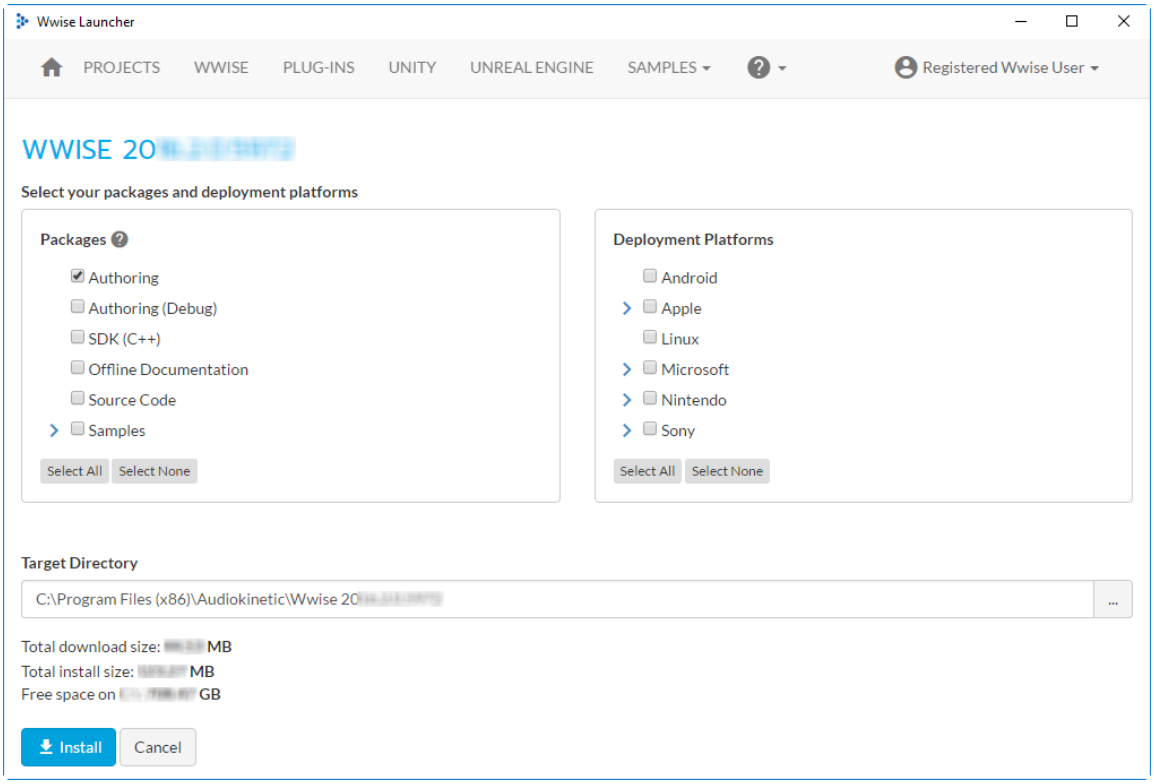
   You'll see a friendly greeting and a first directive to getting started with Wwise.

Tip

If you don't have an account, click the conveniently placed **Sign up now** to create one.



2. Although you can click **Install another version**, the Launcher's big blue install button defaults to the latest available version. Click it to move to the installation screen.



Although the Launcher UI is designed to make Wwise installations intuitively easy, further on this document presents details on how To install Wwise.

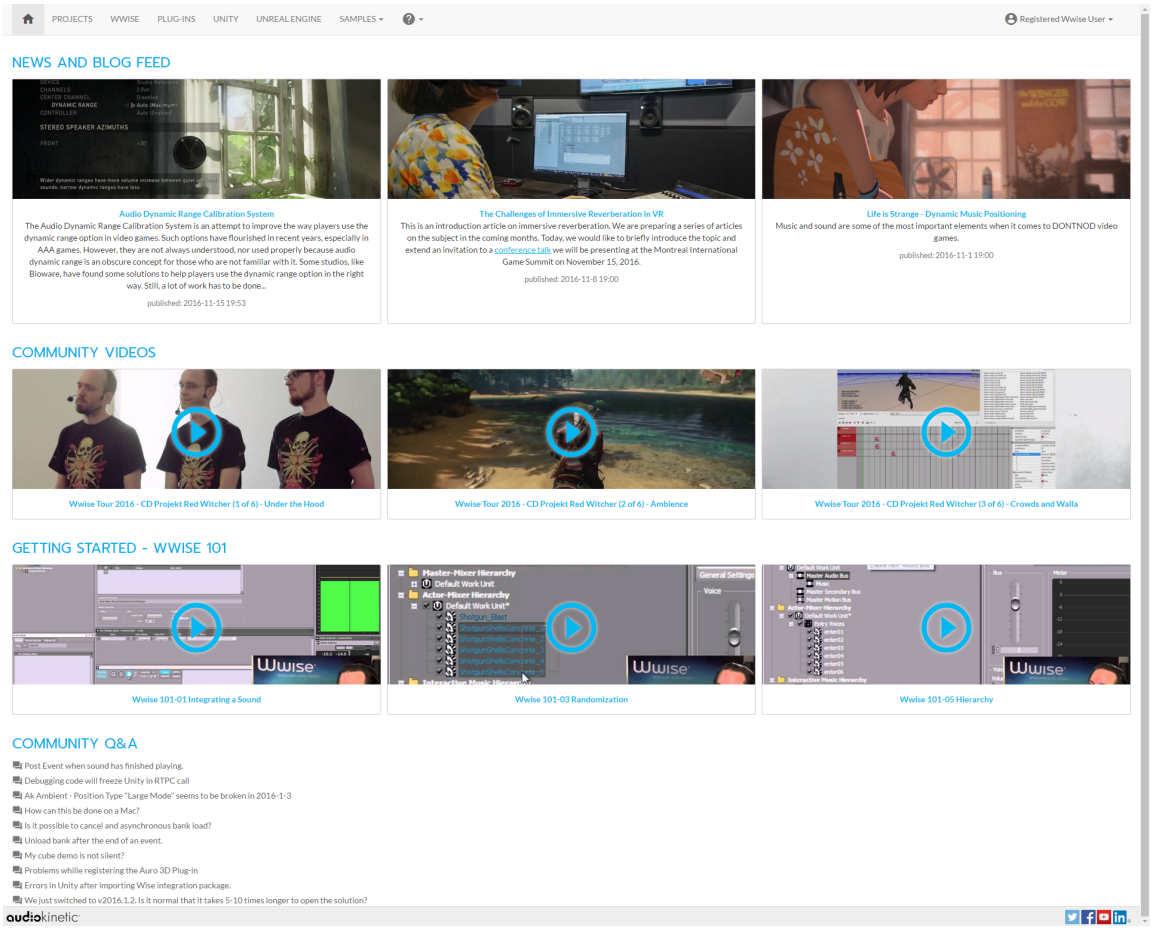
Launcher Overview

The Launcher is designed for easy navigation to key Wwise-related tasks. These are organized into seven distinct pages accessed from their respective tabs: Home, PROJECTS, WWISE, PLUG-INS, UNITY, UNREAL ENGINE, SAMPLES, and Help.

Home

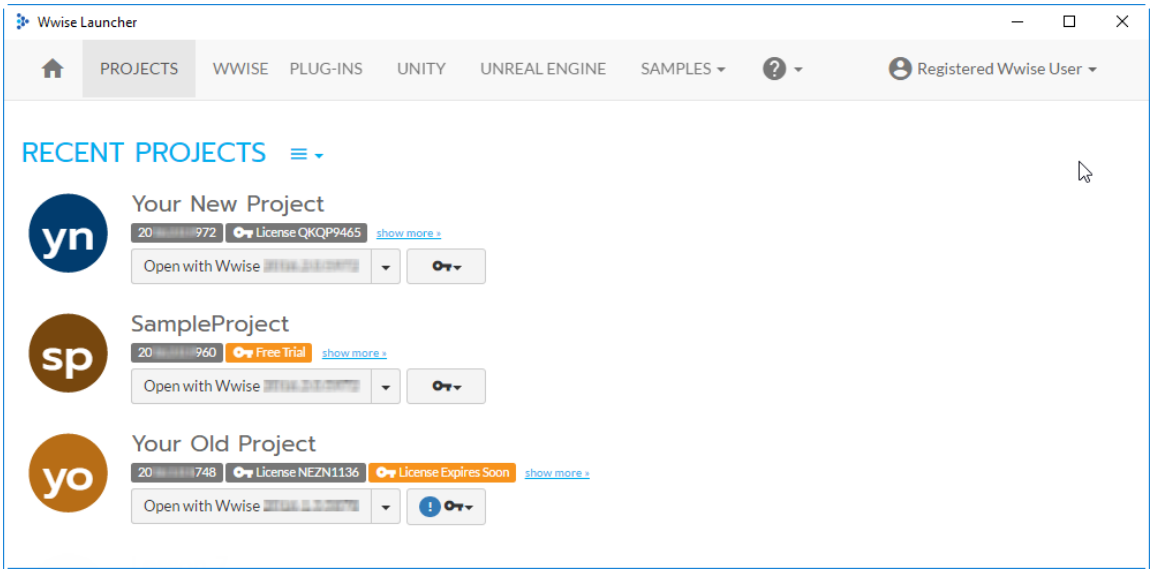
The Launcher opens to its Home tab, seen in the screenshot below. You will see rows for:

- **NEWS AND BLOG FEED**: Lists the latest blogs or news articles on the Audiokinetic website.
- **COMMUNITY VIDEOS**: Lists helpful Wwise user-developed videos posted on YouTube.
- **GETTING STARTED - WWISE 101**: Lists a series of Wwise instructional videos, developed by Robert Brock of the Conservatory of Recording Arts and Sciences as part of the Wwise-101 certification program.
- **COMMUNITY Q&A**: Lists linked titles of the most recent questions posted on the Audiokinetic website's message board, which is designed to allow Wwise users to help each other out.

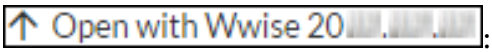


PROJECTS

The PROJECTS tab, seen below, lists in order of recency all the previously loaded Wwise 2016.1 or later projects (including those using the Unity or Unreal integrations). To make it easier to quickly find the project you want, each one is given a permanent identifying icon. With a specific color and letters derived from the project's name (click **show more** to see the full project path), you and your colleagues can quickly jump to different projects across your team's network.



Each project displays an associated version number and licensing information. They also have a list menu with the following options:

- **Open with Wwise ####.#.#.####**: Select to open the project with a specific installed version of Wwise, from among the versions listed in the Wwise tab which are newer than the associated project version. (It's possible to update a project to a newer version, but not possible to open a project with an older version of Wwise.)
  - 

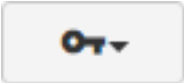
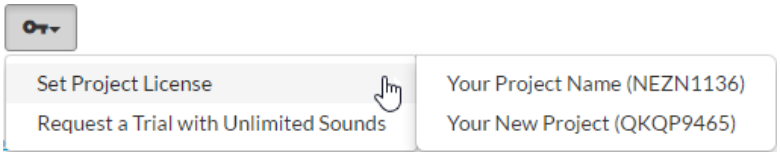
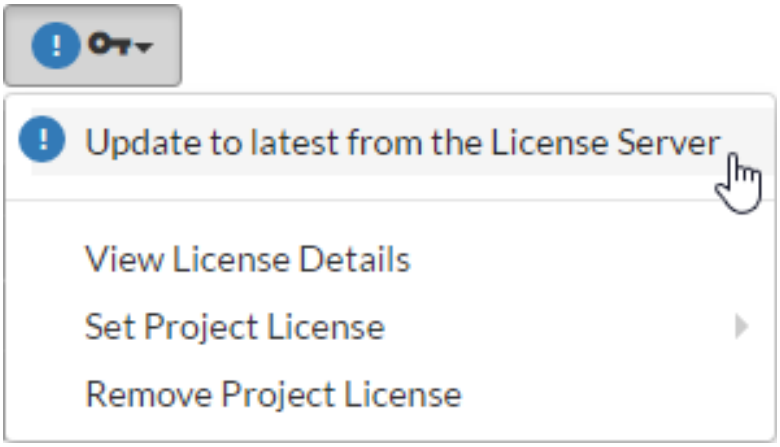
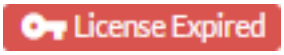
: Migrates the selected project to a more recent version of Wwise.
- **Open Containing Folder**: Select to open Windows Explorer or Mac Finder to the folder where the project's WPROJ file is found.
- **Zip Project...**: Select to prompt the Wwise Project Zipper tool, which you can use to compress a Wwise project and accompanying files. This makes it easier to send a full project to those who may not be on your network, such as Audiokinetic Support.

Note

This option is not available on Mac. Essentially, the manual equivalent of the tool would be to put the project into a compressed folder without the GeneratedSoundBanks folder and without WAV/AMB files from the Originals and .cache folders.

Table 2.1. Licensing information

Display	Meaning
	Indicates the project is in **Trial mode**. By default, all Wwise projects begin with this status. All standard Wwise options are available under this default licensing mode; however, users are limited to a maximum 200 sounds in their SoundBanks, and it is not permitted to publish anything other than non-commercial titles with fewer than 200 sounds. To have these rights, it is necessary to license your project.

Display	Meaning
	This licensing button's menu permits you to set the project's license from among those licenses applicable to your user account. Please make certain to log in to see your available licenses. You can also choose to request a trial license with unlimited sounds.  If a license is applied to the project, there are also options for viewing its details (within the Customer Portal) and for removing the applied license.
	Indicates that something has changed with the currently applied license. For example, a licensed plug-in may have been added, or an expiration date may have been extended. Select **Update to latest from the License Server** to apply the changes. 
	Indicates the license, with its associated project ID, currently set for the project.
	Indicates that the license applied to the project will expire within a week. Your organization's license manager may wish to contact Audiokinetic sales to renew or extend the license.
	Indicates that the license applied to the project has expired. Functionally, this means your project is now in Trial mode. Your organization's license manager may wish to contact Audiokinetic sales to renew or extend the license.

WWISE

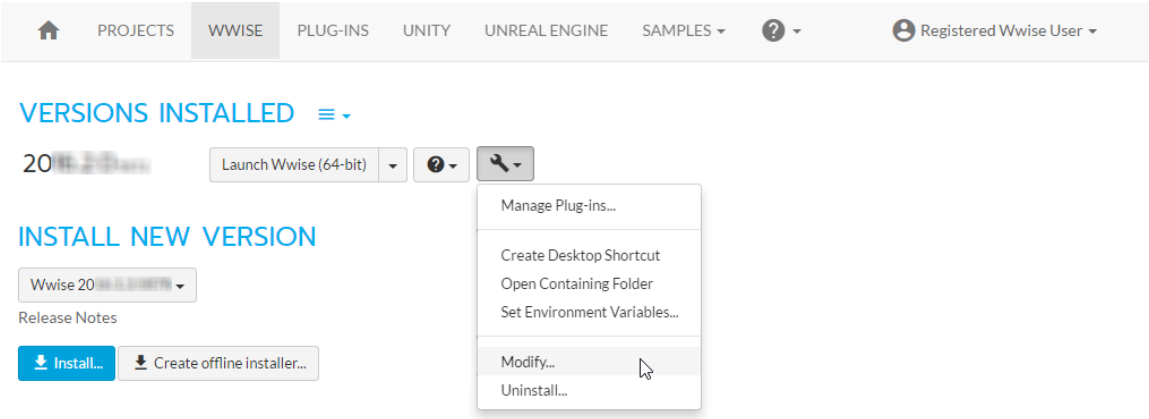
The WWISE tab, seen below, is divided into two sections:

- **Versions Installed**: Lists all the installed versions of Wwise on your machine. Each version has three associated menus:
  - The launch menu, listing options to open Wwise and its associated tools (the File Packager, the Wwise Wave Viewer, and the Multi-Channel Creator).

Note

On Mac, users must open Wwise tools from within Wwise.

  - The documentation menu, listing different installed pieces of documentation, such as the SDK Documentation for different platforms or the User Guide PDF in English, Japanese, or Simplified Chinese.
  - The settings menu, listing options to change your installation, such as managing plug-ins, modifying the installation, or even uninstalling this version of Wwise.
- **Install New Version**: Lists all the recently available versions of Wwise, including both those installed and not installed on your machine.
  For already installed versions, an orange indicator appears to the right. Below the version, three buttons appear for the most commonly needed actions: **Modify**, **Uninstall**, or **Create offline installer...** for this Wwise version.
  For uninstalled versions, only two buttons appear below the version: **Install** and **Create offline installer...** for this Wwise version. See more about installing a new version of Wwise in Installing or Upgrading Wwise and its Packages.



Note

The Launcher works with previously installed versions of Wwise, but only Wwise 2016.1 versions and later are available for install from within the Launcher. Earlier installed versions will be marked as **Legacy**.

PLUG-INS

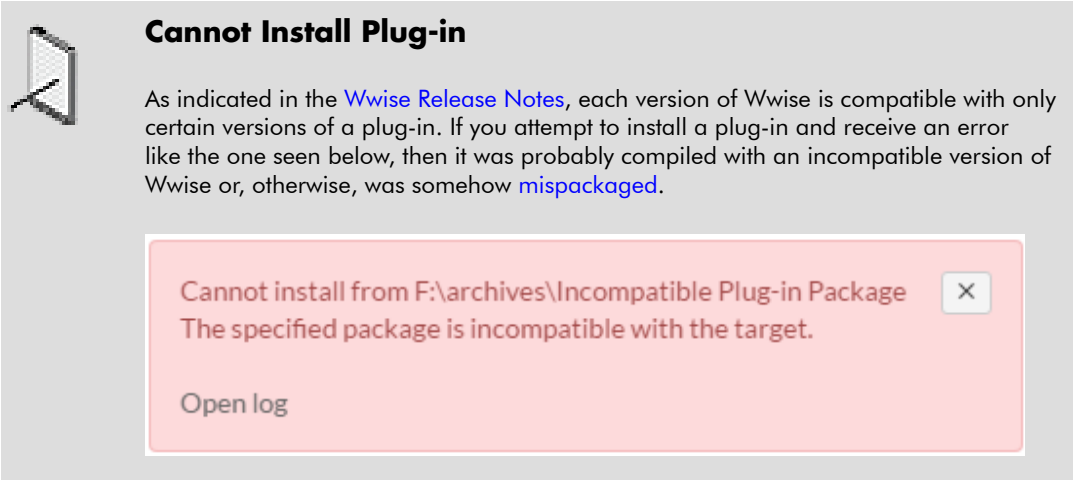
The PLUG-INS tab, seen below, is divided into two sections:

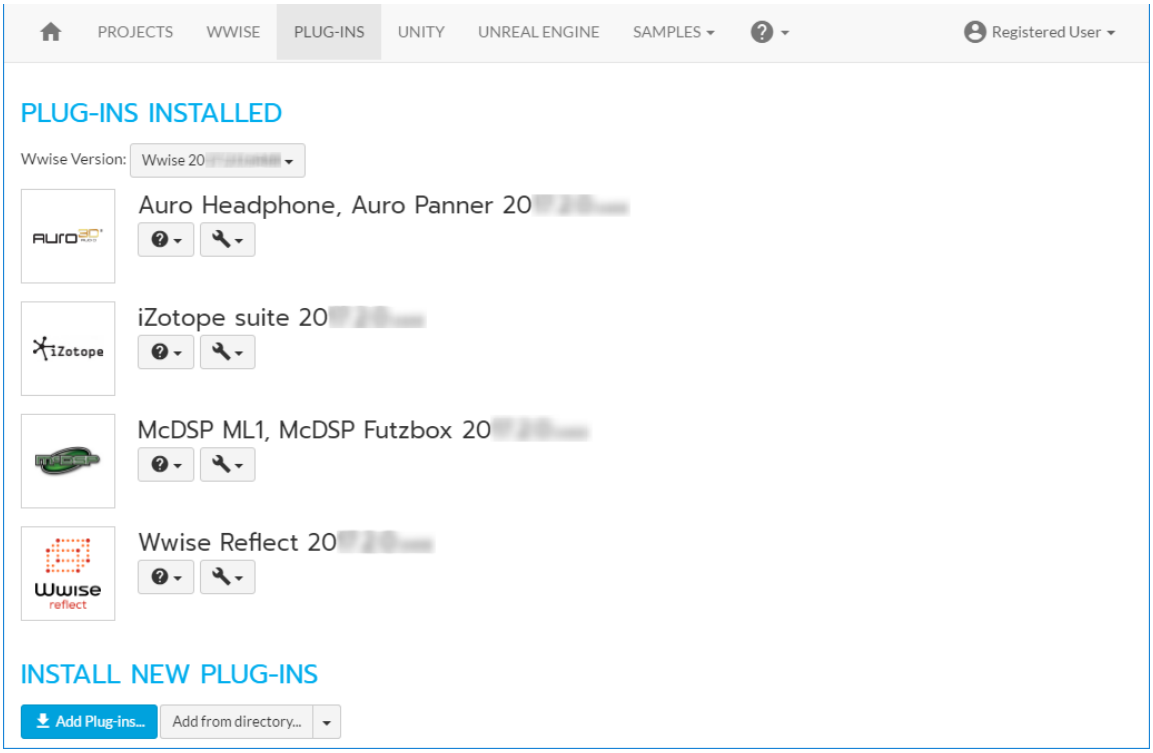
- **PLUG-INS INSTALLED**: Lists all your installed optional plug-ins for the version selected in the **Wwise Version** list. For each plug-in, there are two associated menus:
  - The documentation menu, listing a More Info link to the website overview of the plug-in, the different pieces of plug-in-specific documentation, and all the installed

pieces of documentation for this version of Wwise, such as the SDK Documentation for different platforms or the User Guide PDF in English, Japanese, or Simplified Chinese.

- The settings menu, listing options to **Open Containing Folder** to the Wwise installation directory in Windows Explorer or Mac Finder, or to **Uninstall...** the plug-in (with a confirmation prompt).

- **INSTALL NEW PLUG-INS**: Displays two buttons for adding plug-ins:
  - **Add Plug-ins...**: Click to prompt the CHOOSE PLUG-INS page, where you can install Audiokinetic premium or Audiokinetic partner plug-ins.
  - **Add from directory...** or **Add from archive...** (list selection): Click either of these to open Windows Explorer or Mac Finder and navigate to the location of a 3rd-party (not Audiokinetic) plug-in in order to install it. If the plug-in package is already extracted from its TAR or ZIP archive, use the **Add from directory...** option; if the plug-in package is still in its TAR or ZIP file, use the **Add from archive...** option.

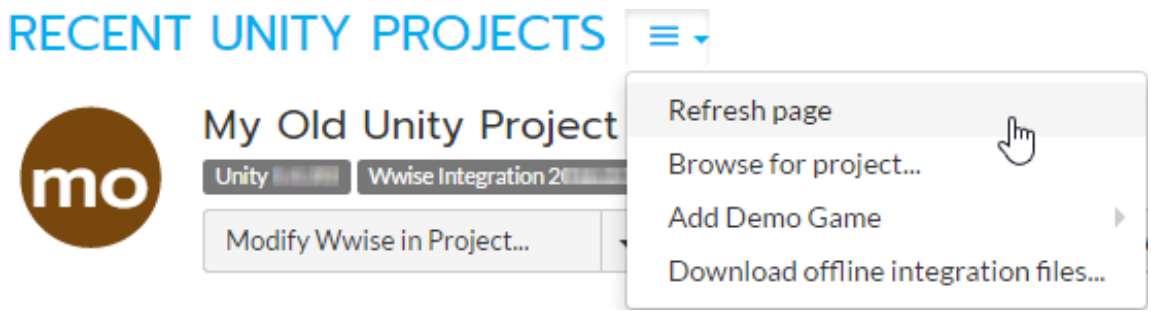




UNITY

The UNITY tab lists Unity's store of most recent projects. If the desired project is not found, you can use the RECENT UNITY PROJECTS menu to:

- **Refresh page**: Prompts the Launcher to look anew for Unity's most recent projects.
- **Browse for project...**: Opens Windows Explorer or Mac Finder to allow you to navigate to the location of your Unity project.
- **Add Demo Game**: Expands a submenu of Wwise versions, which will prompt the Windows Explorer or Mac Finder for you to specify the location to save that version of the WwiseDemoGame. This simple Unity project can serve as a tutorial for the development of your own Wwise Integrated Unity project.
- **Download offline integration files...**: Moves to the DOWNLOAD INTEGRATION FILES page, where you can select the versions for the Wwise Unity Integration, as well as the download directory, before downloading the integration files. With these files, users can then click **Integrate Using Offline Files...** to integrate a Wwise version into a Unity project, without needing an Internet connection.

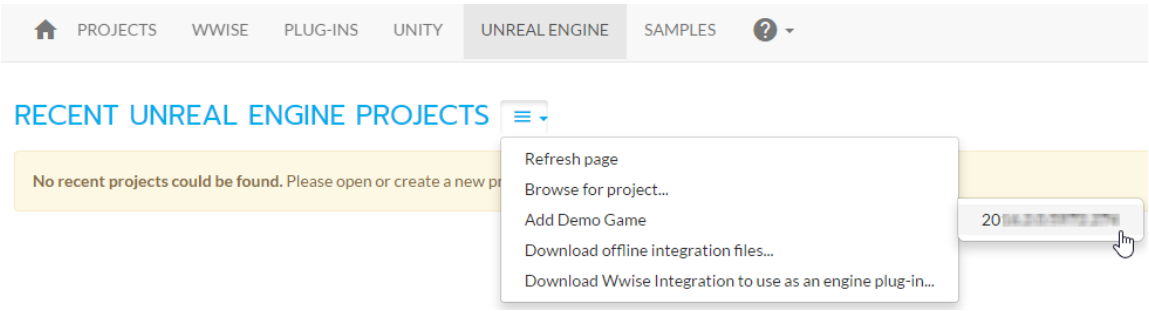


See Integrating Wwise into an Unreal or Unity Project for further details on this tab and the Wwise Unity integration.

UNREAL ENGINE

The UNREAL ENGINE tab lists UNREAL's store of most recent projects. If the desired project is not found, as in the screenshot below, you can use the RECENT UNREAL ENGINE PROJECTS menu to:

- **Refresh page**: Prompts the Launcher to look anew for the Unreal Engine's recent projects.
- **Browse for project...**: Opens Windows Explorer or Mac Finder to allow you to navigate to the location of your UPROJECT file.
- **Add Demo Game**: Expands a submenu of Wwise versions, which will prompt the Windows Explorer or Mac Finder for you to specify the location to save that version of the WwiseDemoGame. This simple Unreal project can serve as a tutorial for the development of your own Wwise Integrated Unreal project.
- **Download offline integration files...**: Moves to the DOWNLOAD INTEGRATION FILES page, where you can select the versions for the Wwise Unreal Integration and Unreal, as well as the download directory, before downloading the integration files. With these files, users can then click **Integrate Using Offline Files...** to integrate a Wwise version into an Unreal project, without needing an Internet connection.
- **Download Wwise Integration to use as an engine plug-in...**: Moves to the DOWNLOAD WWISE UNREAL INTEGRATION page, where you can select the versions for the Wwise Unreal Integration and Unreal, as well as the download directory, before downloading the integration files specific to the engine plug-in. For the expert users who may be able to handle installing the Wwise plug-in as an engine plug-in, there is a conveniently placed link near the bottom of the page to the Wwise Unreal Plug-in > Installation page, which details all the subsequent installation steps.



See Integrating Wwise into an Unreal or Unity Project for further details on this tab and the Wwise Unreal Engine integration.

SAMPLES

The SAMPLES tab, seen below, has a list of three options representing the three sample pages for accessing educational Wwise-related projects:

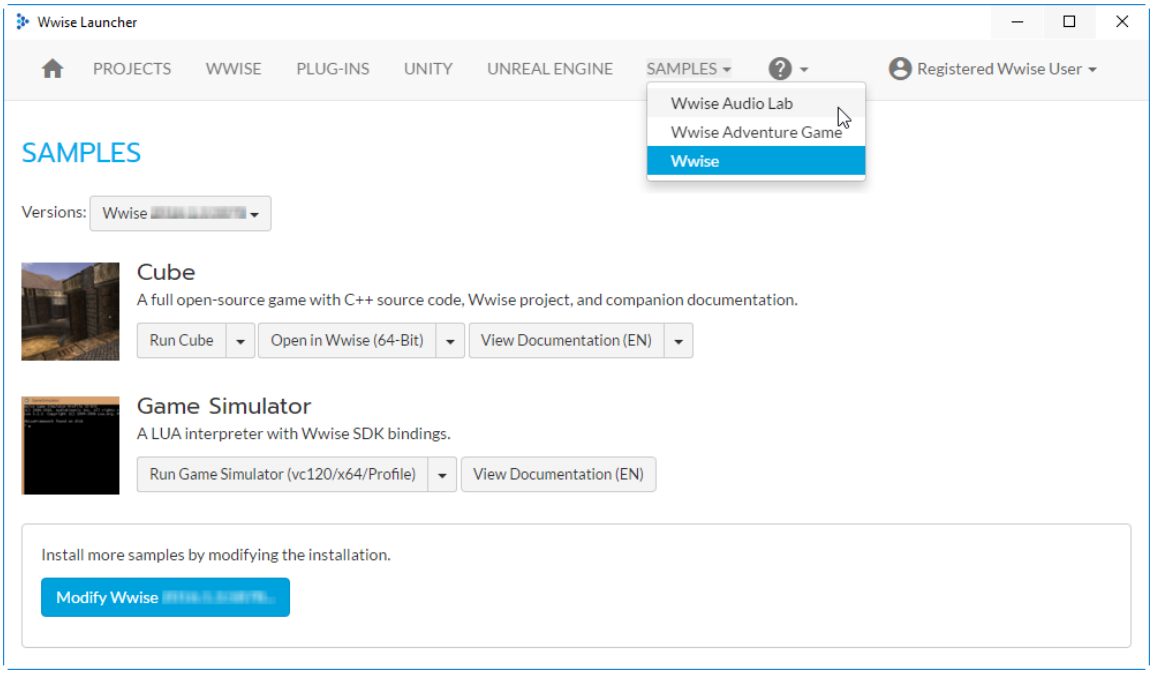
- **Wwise Audio Lab**: Opens the WWISE AUDIO LAB page, where you can install and run the compiled WAL game, its associated Wwise project, and its PDF documentation.

  WAL is an open-source game-like 3D environment developed with Unreal Engine 4. It showcases various spatial audio methods and technologies, which you can learn about in-game with useful information nodes and hands-on experimentation.
- **Wwise Adventure Game**: Opens the WWISE ADVENTURE GAME page, where you can install and run the compiled WAG game, its associated Unity project, its associated Wwise project, and its PDF documentation.

  WAG is a third-person action-adventure game developed with Unity. This is a full game with quests, items to retrieve, enemies to overcome, and a final combat to defeat the plague scourging Allegro Kingdom. It's a great playground to learn and experiment with the Wwise integration for Unity.
- **Wwise**: Opens the main SAMPLES page, where you can open samples installed (see Installing or Upgrading Wwise and its Packages) for the selected version of Wwise.

Tip

Change the tab to list sample projects for another installed Wwise version by selecting from the **Versions** list just below the Samples tab title. If a desired sample is not installed for a particular Wwise version, you can modify that version's installation directly from the Sample tab by clicking **Modify Wwise <version>** below the list of samples.



Each listed sample includes applicable buttons or selectable list options to open documentation and to run Wwise projects, associated games, or small programs such as the Game Simulator LUA interpreter and the Integration Demo.

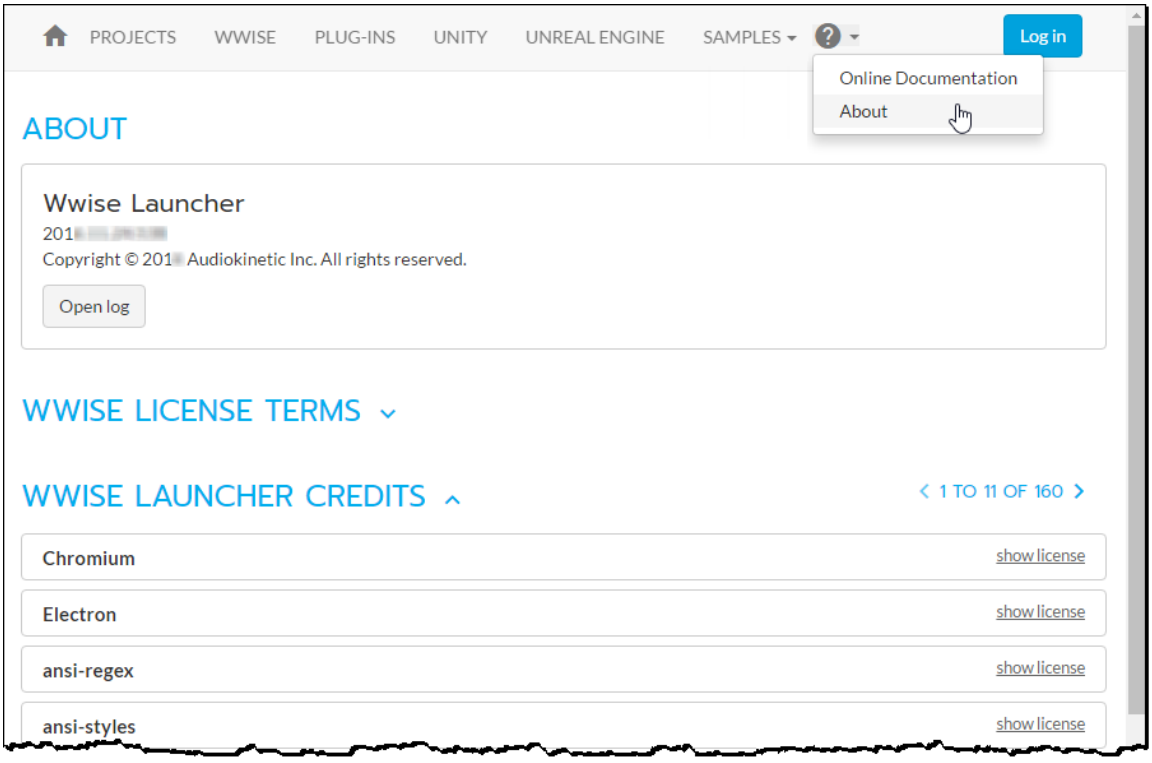
The Integration Demo is an SDK sample

Unlike other samples, which are explicitly selected from among the Samples Packages during Wwise installation, the Integration Demo and the deprecated SoundFrame Application are included as part of the Wwise SDK. By installing the Wwise SDK, you will have these SDK samples.

Help

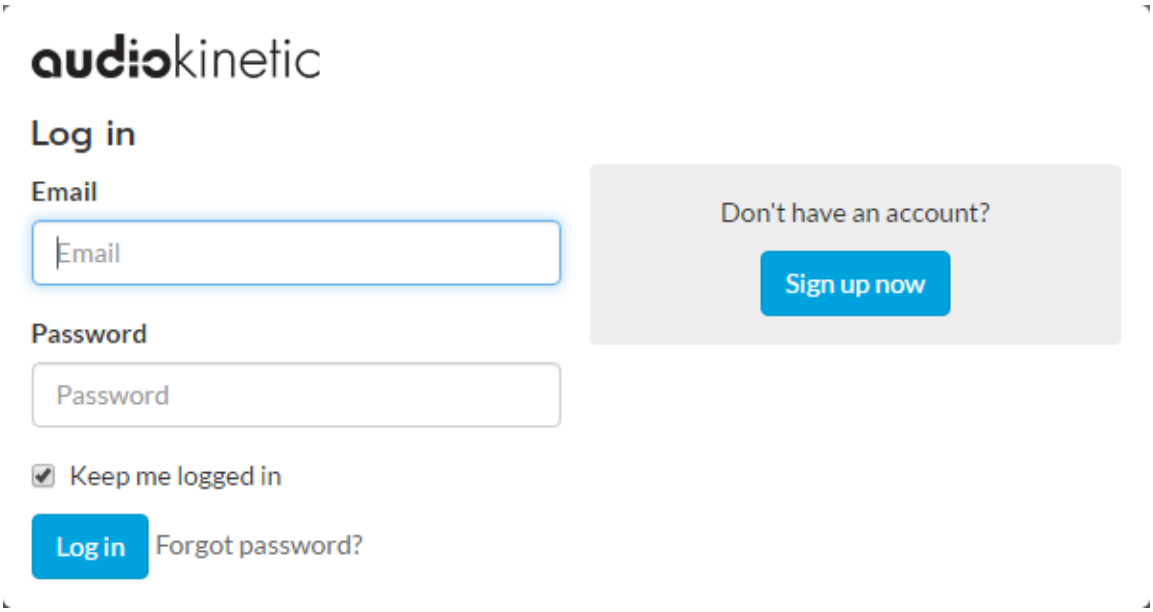
The Help button opens a list of two options:

- **Online Documentation**: Select to open your default web browser to the latest available version of the online Wwise Help.
- **About**: Opens the page to three Wwise information sections:
  - ABOUT: Lists the version number and copyright information of the Wwise Launcher.
  - WWISE LICENSE TERMS: Lists the Wwise license as well as all the licenses of its applicable commercial or open source components.
  - WWISE LAUNCHER CREDITS: Lists all the licenses of the third-party technologies used to develop the Wwise Launcher.



Log in

Click **Log in** to log in to your account and get access to all your customer privileges.



If this is your first time using Wwise or you have a new project to work on, you may want to click **Sign up now** to set up a user account and get access to our Customer Portal.

Installing or Upgrading Wwise and its Packages

The Launcher makes it easy to install Wwise and its component packages.

Be Sure to Install All the Mandatory Code Dependencies

It is not uncommon for a crash to occur due to one or multiple missing dependencies when installing Wwise without the Launcher. The required dependencies are detailed in the following rows of the Wwise System Requirements table: Microsoft .NET Framework, Microsoft DirectX, and Microsoft Visual Studio Redistributable. Installing the Authoring tool through the Launcher is less troublesome because you don't have to worry about the dependencies since they are automatically provided.

There are two primary installation methods:

1. **Using an offline installer.** This method, described in the Working with Offline Installers section, is probably the easiest and most common way to install Wwise while ensuring that everyone on a team of developers or designers has the same version and setup.
2. **Working directly within the Wwise Launcher.** This method, described in the following lines that make up the rest of this section, offers more control over what you decide to install for Wwise. It may be better suited to project administrators.

In the Wwise tab, the INSTALL NEW VERSION block helps you keep track of multiple versions of each Wwise component; the build and version number are prominently displayed.

If you obtain authorization to develop on more platforms, contact sales to change your profile and then you can download and install the new platform files.

To select the installation

1. In The Wwise Launcher, go to the **Wwise** tab.
2. In the **Install New Version** section, select the version you wish to install from the versions list.
3. Click the **Install...** button

To install Wwise

1. In the **Packages** list, select **Authoring** to have the Wwise Authoring Tool.
2. Select the other packages you wish to install.

Tip

For more information about the different packages, click the **?** beside the **Packages** title.

3. Select the platforms on which you will be deploying.

Note

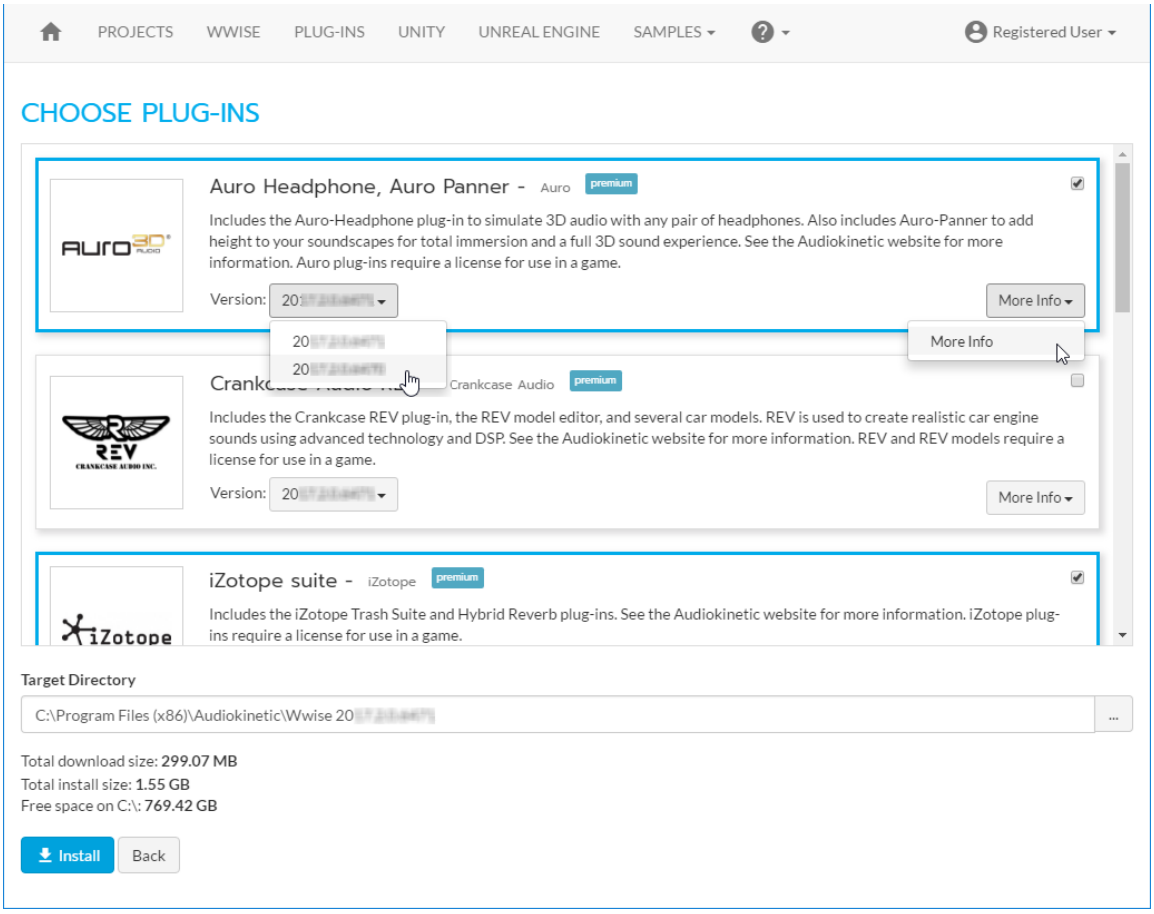
Some check box options may be grayed out. Normally, this is because your user account is not associated with a project that has full rights to use certain licensed products, which can include different deployment platforms.

4. Click **Next**.
You are brought to the CHOOSE PLUG-INS page.

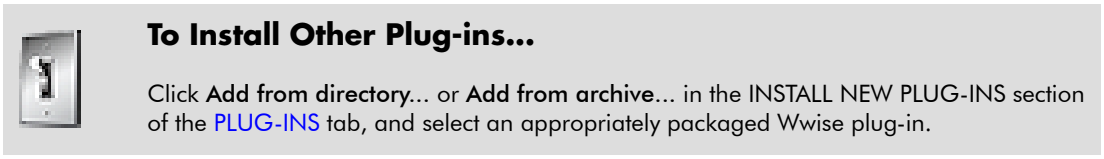
CHOOSE PLUG-INS

This page lists all the optional plug-ins developed by Audiokinetic and its partners. Next to the plug-in name is the developer name. And, if it requires an additional license to be published in-game, a "premium" icon appears next to that. (Plug-ins are free to use within the authoring tool.) Each plug-in includes a description and a version number. There is also a More Info list where you can select:

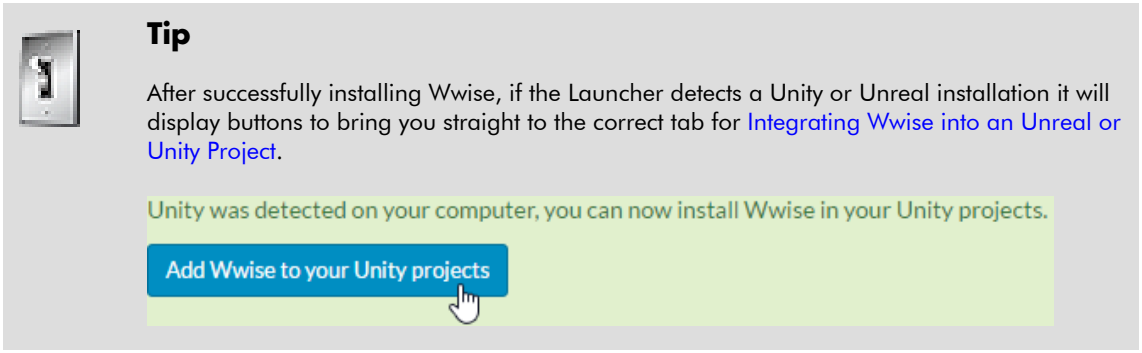
- **More Info** to go to the website overview of the plug-in; or, as applicable,
- **Online Documentation** to go to the online plug-in-specific documentation.



- For all plug-ins you wish to install, click anywhere on the plug-in to enable the check box (upper right corner). (Plug-ins that were already installed will display **Installed** instead of the check box.)
  The plug-in outline turns blue.



5. Click **Install**.
   Wait until the installation completes.



Working with Multiple Versions of the Wwise SDK

When you install a new version of the SDK, the new installation overwrites the WWISESDK environment variable that points to the SDK location. This means that projects from older versions that use this variable will not point to their corresponding SDK version. To solve this issue, you will need to set the environment variable based on the version that you want to use. You could do this in one of the following ways:

- Use the Launcher to set the environment variables for you. This is the newest and easiest method. From the Wwise tab, open the Settings list of the older installed version of Wwise that you wish to use with the SDK, then select **Set Environment Variables...**. A prompt will ask you to confirm the proposed WWISEROOT and WWISESDK environment variable changes to that version of Wwise.
- Use an existing build script to set the path correctly before the build is started.
- Set the environment variable in the command prompt before launching the build tool.

The WWISEROOT environment variable points to the main installer folder (for example c:\Program Files\Audiokinetic\Wwise v2015.1 build XXXX\). It can be used, for example, by tools or scripts that look for WwiseCLI.exe to build SoundBanks.

Integrating Wwise into an Unreal or Unity Project

Your Wwise installation can work within the Unity and Unreal game engines; but, it is necessary to install separate integration packages. Fortunately, the Launcher makes this task very easy.

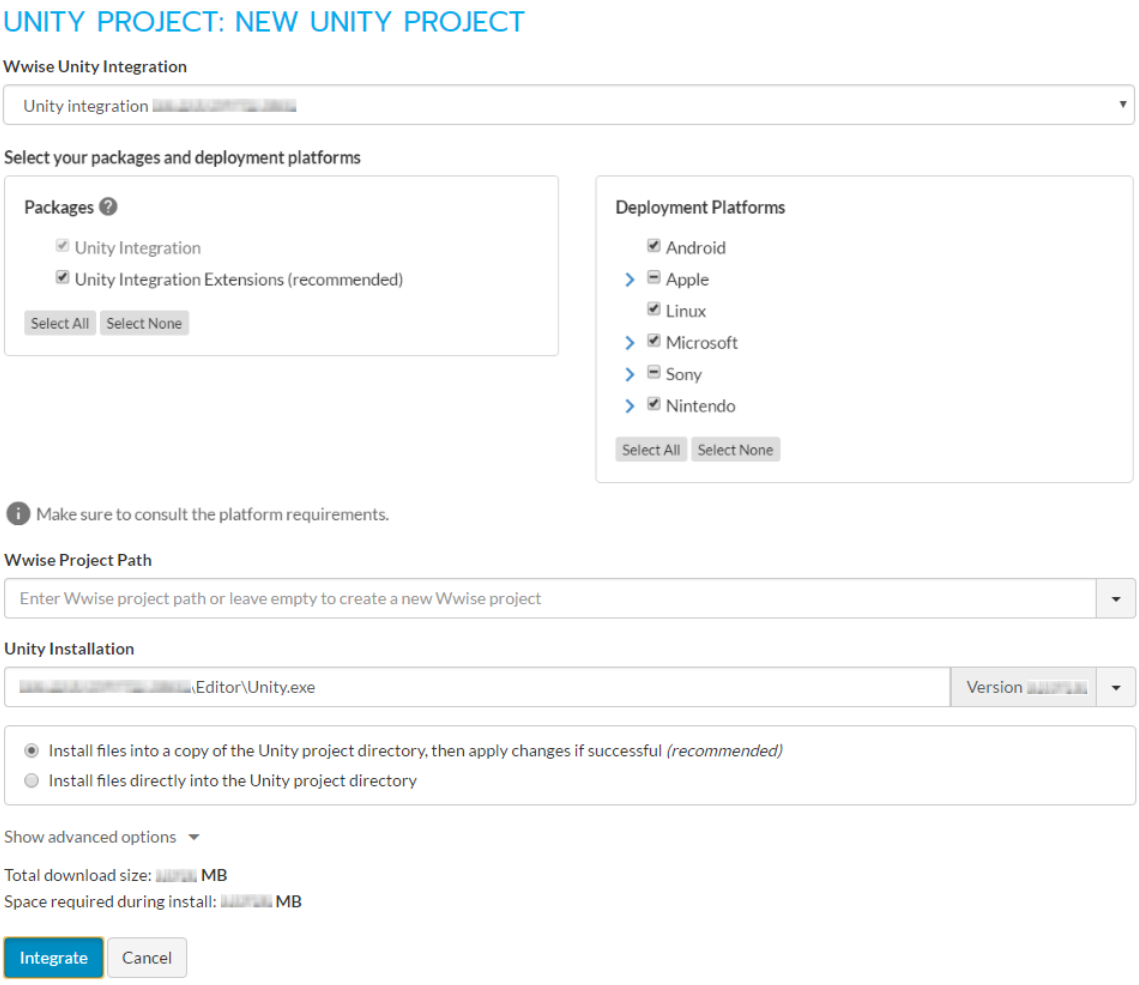
Note

Within the Launcher, only Wwise 2016.2 and later versions can be integrated into Unreal and Unity.

To integrate Wwise into Unity or Unreal projects:

1. In The Wwise Launcher, move to the desired game engine tab, namely UNITY or UNREAL ENGINE.
2. Click the **Integrate Wwise into Project...** button for the desired project.

   The integration page, such as the Unity one seen below, appears.



3. Select the desired Unity/Unreal integration version.
4. **For Unity**: Specify the packages and platforms for this project. The platform-specific source ZIPs will be copied to the Unity project's root folder.

Unity Integration Extensions

Note that the functionality encompassing the auto-registration of `GameObject`'s has been moved to the Unity Integration Extensions. This code can be extended to allow users to modify auto-registration or add more complex `GameObject` handling. For those who know that they do not require this functionality, the Unity Integration Extensions do not need to be installed. Without the Unity Integration Extensions, users are responsible for registering all `GameObject`'s either by attaching `AkGameObj` components or by explicitly calling the API.

For Unreal: Specify the location of your Wwise SDK files, typically found with your Wwise installation.

Note

A note will appear to explain which folders are required and which platform-specific Wwise SDKs will be copied. If needed or desired, you can click **Modify Wwise** `####.#.#.###` to change these settings.

5. Enter the corresponding Wwise project path, or leave the field empty to create a new Wwise project version.
6. **For Unity**: Specify the Unity Installation location for the selected Unity version.
7. **For Unity**: Specify via radio buttons the Unity Installation option you prefer:
   - **Install files into a copy of the Unity project directory, then apply changes if successful (recommended).**

     This is the default installation method. It is secure because in the event where a Unity script won't compile or a file is locked, having installed the files into a copy of your Unity Project enables you to abort the process and/or fix issues without affecting your original Unity project directory.
   - **Install files directly into the Unity project directory.**

     This option is only recommended in a few rare scenarios, such as for users with an enormous project or those installing the integration at a late development stage, if they feel it unnecessary and a waste of precious memory to have a full copy of their project.

Tip

Using a source control management software (such as Git, Subversion, or Perforce) is highly recommended in parallel with the **Install files directly into the Unity project directory** installation option because version control systems enable its user to revert unwanted changes.

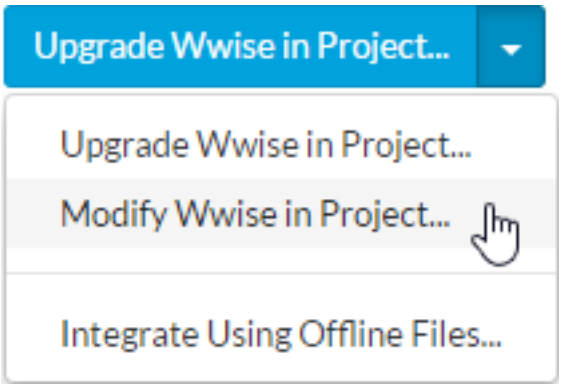
Optionally for advanced users, within your Unity project, you can also have the Launcher create a `WwiseGlobal` GameObject or add the `AkAudioListener` to the MainCamera.

For Unreal: Enter a number between 0 and 4 for the **Max Simultaneous Reverb Volumes**.

8. Click **Integrate** to apply the operation to your project. Click **Cancel** if for some reason, such as the indicated **Total download size** being too great, you do not wish to proceed with the Wwise integration into your Unity project.

To upgrade or modify an integration, select the appropriate project list option:

- **Upgrade Wwise in Project...**: Brings you to essentially the same integration page as for Integrating Wwise into an Unreal or Unity Project; but, the integration version is set to the latest version and the button at the bottom is labeled **Upgrade**.
- **Modify Wwise in Project...**: Brings you to essentially the same integration page as for Integrating Wwise into an Unreal or Unity Project; but, the integration version is set to your current version (with the expectation you will modify some of the other settings) and the button at the bottom is labeled **Modify**.



For further information on installing the...:

- **Unity Wwise Integration**, please see the Wwise Unity Integration > Installing or Upgrading the Integration in a Unity Project documentation.
- **Unreal Wwise Integration**, please see the Wwise Unreal Plug-in > Installation page.

Working with Offline Installers

For both Wwise and Wwise integrations for Unity and Unreal, the Launcher needs to be connected to the Internet to be able to download and then, subsequently, install files. However, for different reasons, users may not always have the necessary connection. Consequently, the Launcher allows users to download offline installers, which users without working connections can use to install Wwise.

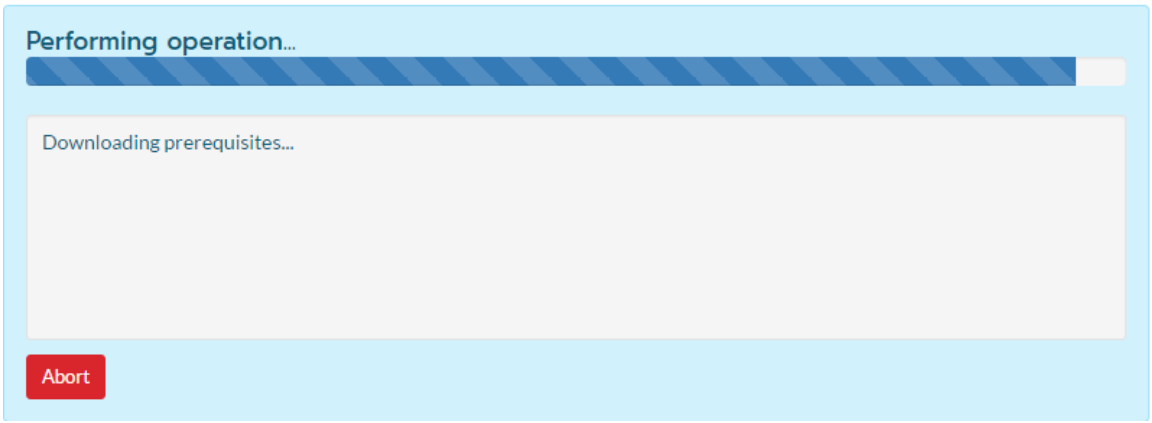
To create an offline installer:

1. In The Wwise Launcher, go to the **Wwise** tab.

2. In the **INSTALL NEW VERSION** section, select the version you wish to install from the versions list.
3. Click the **Create offline installer...** button.

   You are brought to the offline installer page, which lists all the packages and deployment platforms you can select exactly as is done in the installation screen. The only difference with this page is the blue button at the bottom is **Create** instead of **Install**.
4. Follow all the steps given To install Wwise, except click **Create** for step 4.

   Wait for the creation of the offline installer package.



Tip

Be sure to remember the location where you created the offline installer. Although the creation operation will temporarily show the download under the listed installed versions, it's necessary to specify the directory when selecting **Start offline installer...**.

To install using an offline installer:

1. Browse to the directory containing the desired offline installer.

   It should contain a WwiseLauncher.msi file and an accompanying bundle folder.
2. Open the directory and launch the WwiseLauncher.msi.

   A Wwise Launcher offline installer will install and open to the Offline Installation page (almost identical to the standard installation page) with a predefined list of Packages and Deployment Platforms. You may disable the currently enabled ones; but, you will not be able to add or enable those that are not currently part of this offline installer.
3. Click the **Install** button.

Note

If running an offline installer for an already installed version, you will have a **Modify** button instead. The modification applies only when the offline installer includes packages or deployment platforms that have not already been installed for this version.

Wait until the installation completes.

To install using an offline installer within the Launcher:

1. In The Wwise Launcher, go to the **Wwise** tab.
2. From the VERSIONS INSTALLED main menu, select **Start offline installer...**.
3. Browse to the directory containing the desired offline installer.
4. Open the directory and select its child **bundle** folder.
5. Click **Select Folder**.

   The standard installation page appears with a predefined list of Packages and Deployment Platforms. You may disable the currently enabled ones; but, you will not be able to add or enable those that are not currently part of this offline installer.
6. Click the **Install** button.

Note

If running an offline installer for an already installed version, you will have a **Modify** button instead. The modification applies only when the offline installer includes packages or deployment platforms that have not already been installed for this version.

Wait until the installation completes.

Using offline files to install Unity or Unreal integrations

You can also use offline files to integrate Wwise into your Unity or Unreal projects, or to later modify or upgrade them.

To download the integration files:

1. From the UNITY tab or UNREAL ENGINE tab main menus, select **Download offline integration files...**.

   The DOWNLOAD INTEGRATION FILES page appears.
2. Select the Wwise Unity or Unreal Engine Integration version.
3. **For Unity**: Specify the platforms for this project.

   For Unreal: There is nothing to do at this step. However, as you will need the corresponding Wwise platform SDK for each of your project's platforms, you may wish to verify that the applicable version of Wwise has all the necessary platform SDKs.
4. Specify an empty directory into which you will download the offline files.

5. Click **Download**.

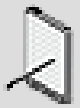
   The Launcher returns to the Unity or Unreal tab where the OFFLINE FILE DOWNLOAD operation runs until completed.

To install offline integration files:

1. In the UNITY tab or UNREAL ENGINE tab, select **Integrate Using Offline Files...** from the first menu of the desired Unity or Unreal project.

   A Windows Explorer or Mac Finder pops up.
2. Browse to the folder with the downloaded offline integration and click **Select**.

   An integration page like the one seen in Integrating Wwise into an Unreal or Unity Project appears. Some of the information is preloaded from the files and cannot be edited, namely:
   - The Wwise Unity or Unreal integration version
   - The mandatory Deployment Platforms for Unity (But the included optional platforms can be removed.)
   - The offline files directory.
3. Otherwise, follow all the normal steps given in the To integrate Wwise into Unity or Unreal projects: list.



Note

Expert users who wish to install the Wwise plug-in for Unreal as an engine plug-in should consult the Wwise Unreal Plug-in > Installation page.

Updating your Version of Wwise

You can upgrade to a newer version of Wwise, yet still maintain previous versions of Wwise on your workstation. This can be very convenient in the following situations:

- You have been working on a project in a previous version, and you want to download the latest version to test out the new features without migrating your project.
- You are working on several projects that use different versions of Wwise. For example, you are:
  - A content provider whose clients are using different versions of Wwise; or
  - An audio designer working on a new project that is using the latest version and wrapping up an older project that is using an older version of Wwise.

All installed versions of Wwise display in the Wwise tab of the Launcher, listed in descending order with clearly marked version numbers. Apart from legacy versions, you will not find shortcuts to Wwise from the Windows Start menu or the Mac Startup menu. For legacy versions, they are also accessible from the Start/Startup menu, where they are categorized by version number.

Remember to open older projects in their corresponding version of Wwise to avoid being prompted to migrate them to the latest version.

It is a good idea to review the Release Notes, especially the Important Migration Notes page, for the current version of Wwise prior to installation so that you are aware of any features, issues, and workarounds that might affect the projects you are currently working on.

The Launcher installs all the necessary components, including the Microsoft® .NET Framework v4.0, needed for the application to work properly. Keep in mind that your installation will only include the platforms you selected to install and for which you are authorized. If you obtain authorization to develop on more platforms, contact sales to change your profile. Then you will be able to download and add the new platform files.

The installation process includes the following steps:

- Wwise System Requirements.
- Installing or Upgrading Wwise and its Packages.

Note

You can uninstall Wwise at any time, if needed. For more information on uninstalling Wwise, refer to Uninstalling Wwise.

Adding an Existing Installation to the Wwise Launcher

Normally, the Launcher should detect all versions of Wwise installed on your system, which it then lists in the WWISE tab. However, it may occur that versions, such as very old ones or possibly those saved under version control, are not detected by the Launcher. In these cases, you can:

- Select **Locate local copy of Wwise...** from the VERSIONS INSTALLED menu.

  A Windows Explorer or Mac Finder will pop up.
- Navigate to the location of the desired Wwise version's wwise.exe file.
- Select the wwise.exe and click **Open**.

  The Explorer/Finder closes and the Launcher returns to the Wwise tab where that version of Wwise is now listed among the VERSIONS INSTALLED.

Note

Although versions previous to 2016.1 can be uninstalled directly from within the Launcher, they cannot be modified.

Adding Platform Components to the Current Installation

If, for example, you become an authorized developer for a new game platform after installing Wwise, you get a new license with access to other plug-ins, or you decide to

add another package, then the Launcher allows you to easily update an already installed version of Wwise.

To add components to your current installation:

1. In the WWISE tab, find the version of Wwise you wish to update from either the listed VERSIONS INSTALLED or the INSTALL NEW VERSION list.
2. **VERSIONS INSTALLED**: In the settings menu, select **Modify...**.

   INSTALL NEW VERSION: Click **Modify...** where previously you would have pressed **Install...** for a new version.

   You'll be brought to the same page as when you needed To install Wwise.
3. Although you won't be able to remove already installed packages or platforms, add new ones as desired and for which you have the appropriate rights.
4. Click **Modify**.

   The operation runs to add the newly selected components.

Adding or Removing Plug-ins

While the Modify option will take you through the CHOOSE PLUG-INS process like an initial Wwise installation, you can more easily add and also remove plug-ins from your installed Wwise versions directly in the PLUG-INS tab.

Uninstalling Wwise

You can uninstall any version of Wwise, whenever you want, from within the Launcher. While it is not necessary to uninstall Wwise before you update to a newer version, you might want to uninstall the previous version if you have no reason to keep it.

Note

In addition to the simplified uninstall option of the Launcher, it is still possible to uninstall legacy versions of Wwise using the standard Windows Control Panel option or the Mac's trash. Current versions, however, can only be uninstalled from the Launcher.

To uninstall a version of Wwise:

- Within the Wwise tab of the Launcher, go to the installed version of Wwise that you wish to uninstall.
- Open the Settings list and select **Uninstall...**.

  A confirmation dialog is prompted.
- Confirm that you want to completely remove this version of Wwise from your computer by clicking **Yes**.

The Launcher will display a blue box, like the one displayed below, indicating the application removal progress. Depending on the size of the selected installation and your machine's power, this may take a few seconds or several minutes to complete.

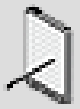
Note

You may notice that some temporary files created in the Wwise and Wwise SDK installation folders still remain after uninstalling Wwise. You can delete these files manually as required.

Chapter 3. Wwise Versioning and Project Upgrades

Wwise Versioning

The versioning system used by Wwise is of the form "YEAR.MAJOR.MINOR.BUILD". Depending on the types of changes in Wwise from one version to another, your projects may not need to be migrated when upgrading to a new version. This is indicated by two parts of the version: the YEAR.MAJOR and the MINOR version.



Note

- (YEAR.MAJOR) Major Release: A difference in the major version indicates that there may be major breaking changes such as removal and changes to APIs, as well as the introduction of major new features and rewrites. Backward compatibility, therefore, cannot be assumed. The fact that the year is present is only for indicative purposes and making reading versions easier than an incremental value. Indexing for the MAJOR part starts at 1, as in 2018.1, 2018.2, 2019.1, and so on. When opening a project saved in a previous major version of Wwise, it will be automatically updated to the newer version. After that, it will no longer be possible to open that project with a previous major version.
- (MINOR) Minor Release: A difference in the minor version indicates that the versions are interchangeable. It is guaranteed that the API remains fully backward compatible. Changes in minor versions include bugfixes and minor API additions. Indexing for the MINOR part starts at 0, as in 2018.1.0, 2018.1.1, 2018.1.2, and so on. No migration is necessary when opening a project saved in the same major version.

We recommend that you **always update to the latest minor version**. This is always the most stable version since it will contain all existing bugfixes for the corresponding major version. It will also not break your usage of the Wwise SDK and Wwise authoring application (including the Wwise Authoring API). Support for newer platform SDKs are also provided in minor versions, so it will likely be needed for shipping a product for these platforms. The various formats used by Wwise will also remain the same across minor versions. You can expect all of the following to remain compatible:

- Soundbanks (no need to regenerate or redeploy)
- Work Unit files
- Wwise SDK API
- Wwise Authoring API (WAAPI)
- Plug-ins

Because products sharing the same major versions are interchangeable, it is possible to profile a game built against a different minor version of the SDK with the Wwise authoring application. Similarly, banks generated with a different minor version, be it more recent or older, will be compatible within the same major version.

Project Upgrades

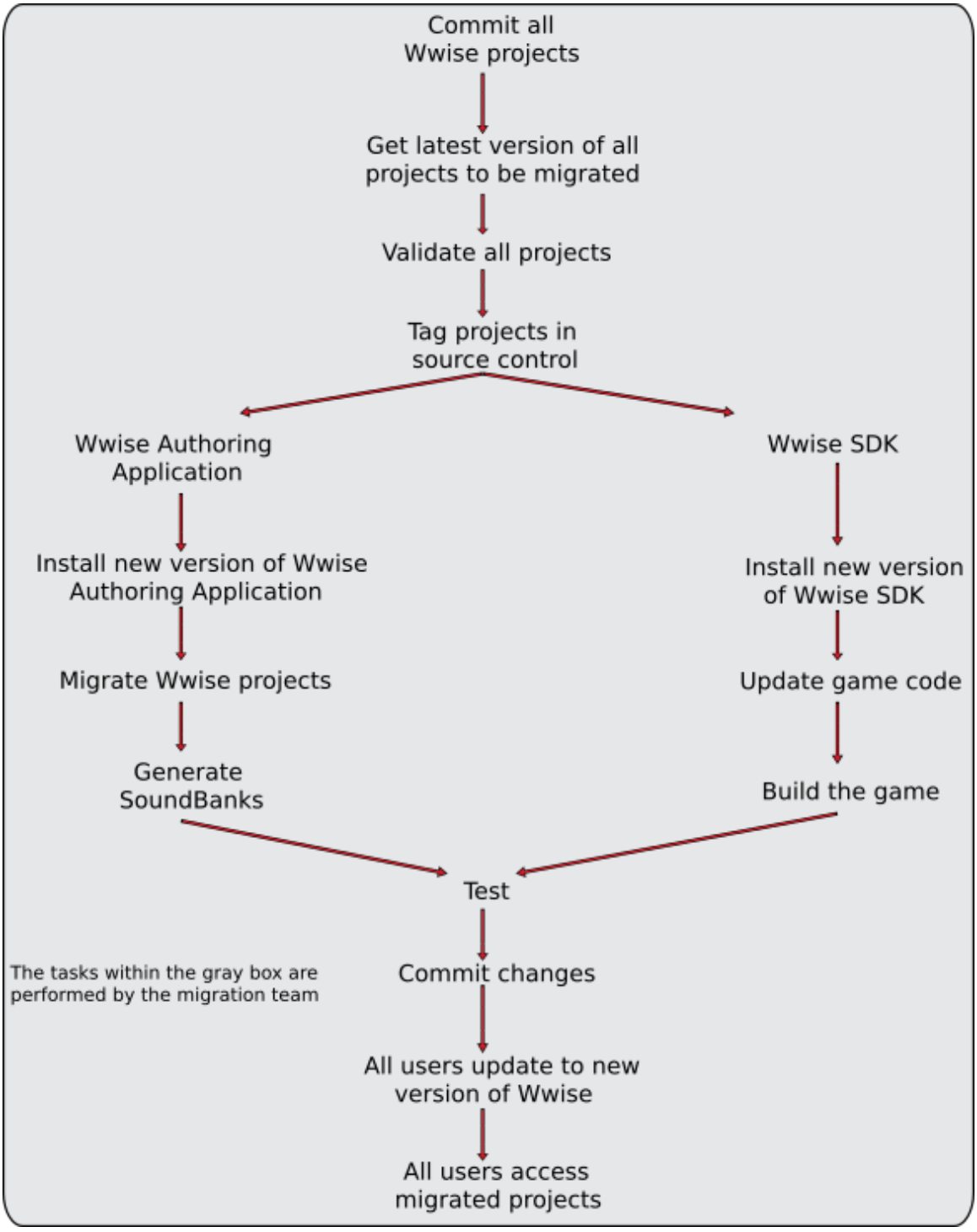
When you are ready to upgrade to a new major version of Wwise, you need to follow a coordinated protocol to ensure that your projects created in the previous version are migrated smoothly to the newer version. The goal is to avoid putting any aspect of your game at risk, so you need to examine the changes in the new version and selectively and systematically implement any modifications to your game code.

The protocol for upgrade and migration may vary depending on the number and configuration of users working with Wwise and the following conditions:

- Projects under source control: The project will need to be checked in at key points.
- Multiple users: The upgrade and the project migration need to be completed for the project and the SDK before multiple users can all upgrade on their workstations.
- Integrated game project: The game code needs to be modified for the changes in the upgrade.

The migration team, composed of one or two designated individuals, should first upgrade and migrate to a local system for the authoring application, and simultaneously upgrade and migrate to a local system for the SDK and the game code as needed. After the team has carried out their respective tasks, all other users can safely upgrade their Wwise installations and access the migrated projects.

For an overview of the upgrade and migration workflow for a project under source control, consult the following chart:



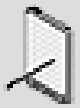
Before You Begin

Before you upgrade to a new version of Wwise, you need to prepare your projects for the upgrade, particularly if your Wwise project has been integrated into a game. Repeat the procedures in the following sections for all projects that you want to migrate.

The migration team should carry out the following tasks to prepare projects to be migrated to the latest version.

1. Commit all projects:
   a. For projects under source control, ask the other members of your team to commit all their changes.

      This will avoid merge problems later.
2. Get the latest version of all projects to be migrated:
   a. For projects under source control, import all the projects locally to your system.
   b. Ensure that all project files have read and write permissions.



Note

For projects under source control, you may need to check out the project file (WPROJ) and all Work Unit files (WWU) from the various subfolders.

3. Validate projects to be migrated:
   a. Open each Wwise project that you plan to migrate in the current version of Wwise, and save.
   b. If the Project Load Log dialog box displays messages and suggested fixes, accept these fixes, save the changes, and close the project. Repeat this procedure until there are no such messages. Your goal is to remove any project errors to simplify the upgrade.
4. Tag projects to be migrated:
   a. For projects under source control, commit and tag or label this as the last version of the project prior to the upgrade.

      Committing the project at this point will record all the changes up to the migration process. Any other changes will be directly associated with the migration in the source control history.

Upgrading Wwise

The project migration process has been designed to ensure that your migrated project very closely approximates the sound of your previous project despite the significant changes made to the feature set. For more information about these changes, refer to the current version of the Wwise release notes.

The migration team should migrate the Wwise project to the new Wwise version on a local system first. All other users should not alter the project until the migrated project has been tested and then committed by the migration team.

Wwise Versioning

Depending on the types of changes made to Wwise from one version to another, your projects may not need to be migrated when upgrading to a new version.

- **Major Release**: If the year number of the version or the first decimal point after the year changes, this is a major release likely involving several major changes. For example, 2016.1 and 2016.2 were both major releases. When opening a project saved in the former version with the latter version of Wwise, Wwise will automatically update it. After that, it will no longer be possible to open that project with the 2016.1 version.
- **Minor Release**: If the year number of the version or the first decimal point after the year remain the same, but there is iteration of an additional decimal point, then this is a minor release. For example, 2016.1.3 and 2016.1.4 were both minor releases with only a few changes. When opening a project saved in the former version with the latter version of Wwise, no migration is necessary.

1. Upgrade Wwise:
   a. Upgrade Wwise on your local system, as described in Installing or Upgrading Wwise and its Packages.
2. Migrate a project to the new version:
   a. Verify that all files in the current Wwise project have read and write status.
   b. Perform a backup of the project and store the backed up project in another folder on your system or network. The project will be backed up automatically by Wwise, but it is a good practice to do your own backup as well.
   c. Open the Wwise project that you plan to migrate to the new version.

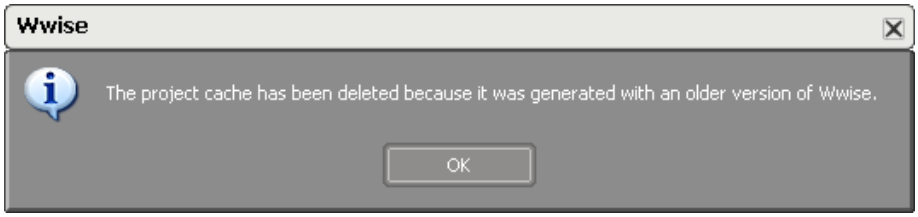
      A message is displayed prompting you to migrate this project to the latest version installed on your system.

Note

The Migration dialog box is only displayed when significant changes have been made to the project file between versions. If your project does not require migrating, Wwise will skip to the next step in the migration process.

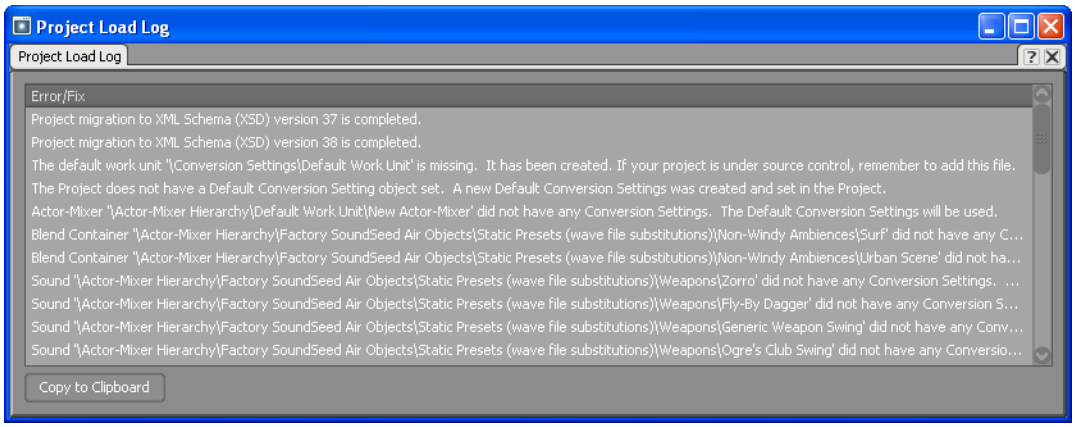
   d. Click **Yes.**

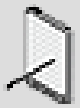
      Wwise begins to migrate the project. After the project is migrated, Wwise checks to see if the existing project cache folder is compatible with the current version of Wwise. If not, Wwise deletes the project cache folder. When the project cache is deleted, the following message box is displayed.



e. Click **OK**.

The Project Load Log dialog box will open if there is migration information and messages to display. The migration information includes the creation of any new files that are required for the new version of Wwise.





Note

This information is stored in the file named projectname.wproj_migration.log in the project folder. If needed, you can also click **Copy to Clipboard** to save this information for reference.

f. If you are using source control, add any new files that were created during the migration process to your source control system.

g. Close the Project Load Log dialog box.

3. Generate SoundBanks:

a. Generate SoundBanks for migrated projects that will be used for testing.

Upgrading the Wwise SDK

Just as you need to prepare to upgrade the Wwise authoring application, you also need to carry out the following tasks to upgrade the Wwise SDK.

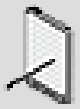
1. Upgrade the Wwise SDK:

a. Consult the Wwise SDK documentation to verify that your current system matches that described in the SDK System Requirements section.

b. Optionally, uninstall the previous version of the Wwise SDK.

It is a good practice to also manually remove any temporary files created by the sample C++ projects.

c. Install the current version of the Wwise SDK.

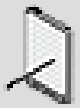


Note

For more information about installing and uninstalling the Wwise SDK, refer to the release notes.

d. Verify that the WWISESDK environment variable points to the installation folder for this current version. (See Working with Multiple Versions of the Wwise SDK for more information on setting the environment variable.)

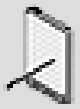
2. Update the game code for the newer version of the Wwise SDK:
   a. To migrate the game code, you will need to port the existing functionalities of a C ++ project to the latest version of the Wwise SDK from the previous version. This task may include modifying or refactoring code in different modules. Ideally, the migration team will work concurrently on updating the game code and migrating the Wwise project to the new version. While the Wwise projects are being upgraded and migrated, changes to the following modules may need to be addressed to update the game code:
      - SoundFrame
      - Sound Engine
      - Memory Manager
      - Stream Manager
      - Communications Module



Note

To ensure that your game is not adversely affected by some of the new features in this version, it is always a good idea to first upgrade the components that you need for your game. After you are satisfied that your game is stable, you can integrate the new features, if needed.

   b. Refer to the changes listed in the Wwise SDK documentation to familiarize yourself with the modifications you will be making.
   c. Modify the code as required.



Note

It is a good practice to rebuild the C++ project often when you are modifying the code.

3. Build your game.

Testing

Before you can complete the upgrade, you need to test the migrated game. The tests are carried out both in the Wwise authoring application and in the game by the migration team. Most likely you have created a testing protocol at your workplace. To augment what

you are already using, we have included a simple procedure that you can consider when you are testing the upgrade.

Test the migrated game project, then commit:

1. In the game, verify the sound and audio behavior. This can include verifying the following:
   - Basic audio playback.
   - Behaviors affected by changes you have made to the code.
   - Platform behaviors.
   - Communication with the Wwise authoring application.
   - Any in-house tools for SoundFrame that might be affected by this upgrade.
2. Commit your migrated Wwise project and the code changes to source control.

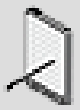
After the testing is complete and you are satisfied with the results, you are ready to instruct other users to upgrade to the latest version of Wwise.

Upgrading Wwise on all Workstations

After the Wwise project has successfully been upgraded and migrated on a local system, the SDK and other code changes have been made, and the testing completed, you are ready to advise other users to upgrade Wwise on their workstations.

This upgrade must be carried out on all the workstations where Wwise is being used, including the following:

- Build machines
- Sound designers' workstations
- Developers' workstations
- Any other Wwise user workstations



Note

Ensure that you have identified all the workstations that run Wwise so that you can upgrade them all.

Upgrading User Workstations

After the development and migration tasks are completed, all Wwise users can upgrade to the current version of Wwise.

Upgrade Wwise on all workstations:

1. Optionally, uninstall all Wwise components. This may include:
   - Wwise authoring application
   - Wwise SDK

For more information about uninstalling, refer to Uninstalling Wwise.

Note

It is not necessary to uninstall Wwise. If required, you can maintain several builds and versions of Wwise on the same workstation.

2. Install the current version of Wwise. This may include:
   - Wwise authoring application
   - Wwise SDK
   - Wwise Game Simulator

   For more information about installing, refer to Installing or Upgrading Wwise and its Packages.

All users access the migrated Wwise project:

1. After you have installed the latest version of Wwise, retrieve the migrated Wwise project.
2. Open Wwise.

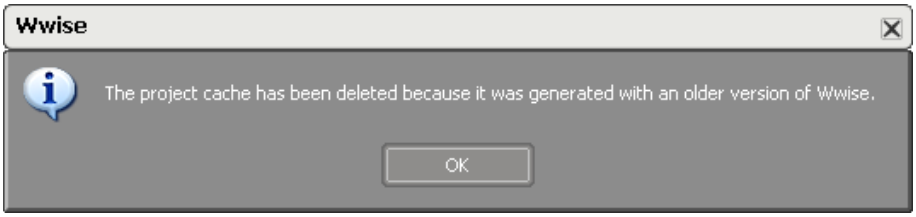
   The End-User License Agreement is displayed.
3. Read the agreement and if you accept, click **Accept**.

   The Project Launcher window opens.
4. Open the migrated project.

Note

If some users have work units in the Wwise project that have been saved locally and are not under source control, a message appears prompting them to accept that these work units will be migrated when the project opens in the new version of Wwise.

Before the project can be loaded, Wwise checks to make sure that the project cache folder is compatible with the current version. If not, Wwise deletes the project cache folder. When the project cache is deleted, the following message box is displayed.



5. Click **OK**.

   After the cache is deleted, the project is loaded into Wwise.